MUD (D)TLS profiles for IoT devices

draft-reddy-opsawg-mud-tls-01
IETF 106, Singapore
Nov 2019

T. Reddy (McAfee)
D.Wing (Citrix)
Agenda

• Recap
  ▪ TLS handshake inspection
  ▪ Observable (D)TLS profile parameters
• Solution overview with examples
• Questions & Comments
TLS handshake inspection

- Detect malware families based on TLS profile.
- Certificate
  - Mismatch between SNI and DNS names in the SubjectAltName(SAN) X.509 extension
  - Self-signed
  - Expired
- Cryptographic parameters
  - Older and weaker cryptographic parameters (e.g., TLS_ECDH_ECDSA_WITH_RC4_128_SHA)
- TLS extensions
  - Low diversity of TLS extensions
  - Extensions used by IoT devices not supported by malware (e.g., Grease)
- Weird hostnames
  - DGA characteristics of SNI and SubjectAltName
- Prevent attacks at TLS layer (expired certificate, weak encryption etc.)
Observable (D)TLS profile parameters

- Useful for IoT devices that very broad communication patterns.
- IoT devices vulnerable to MiTM attacks
- IoT devices can learn new skill and the new skill changes the way IoT device communicates with other devices.
Observable (D)TLS profile parameters

- We profiled several IoT devices: Amazon Echo, Echo dot, Echo Show, Fire TV, Google Home Mini, Google Home and Kindle.
  - Observable (D)TLS profile parameters did not change after learning new skills.
  - (D)TLS profiles for IoT devices based on type, manufacturer and model is also different
- We also observed TLS profile parameters of thousands of malware flows.
Solution overview

- Extends MUD to model observable (D)TLS profile parameters

module: reddy-opsawg-mud-tls-profile
  augment /mud:mud/mud:from-device-policy:
    +--rw client-profile
      +--rw tls-profiles* [profile-name]
        +--rw profile-name                     string
        +--rw protocol-version?                uint16
        +--rw supported_versions*              uint16
        +--rw grease_extension?                boolean
        +--rw encryption-algorithms*           encryption-algorithm
        +--rw compression-methods*             compression-method
        +--rw extension-types*                 extension-type
        +--rw acceptlist-ta-certs*             ct:trust-anchor-cert-cms
        +--rw SPKI-pin-sets*                   SPKI-pin-set
        +--rw SPKI-hash-algorithm              ct:hash-algorithm-t
        +--rw psk-key-exchange-modes*          psk-key-exchange-mode
        +--rw supported-groups*                supported-group
        +--rw signature-algorithms*            signature-algorithm
        +--rw client-public-keys
          |   +--rw key-exchange-algorithms*     key-exchange-algorithm
          |   +--rw client-public-key-lengths*   client-public-key-length
        +--rw actions
          +--rw forwarding                    identityref
Amazon Echo Show

No of Extensions offered

[Bar chart showing the number of extensions offered for both echoShow and malware with specific values at various points.]
Amazon Echo Show

Grease Values

- echoShow
- malware
Amazon Echo Show

Supported Groups

- echoShow
- malware
Amazon Echo Show

Self_Signed

echoShow

malware

Blue: malware
Red: echoShow
Amazon Echo Show

Signature Algorithms

Graph showing data for Amazon Echo Show and malware.
Amazon Echo Show

Cipher Suites

- echoShow
- Malware
Google Home

Cipher Suites

Ghome
Malware
• Observed (D)TLS profile from several IoT devices and thousands of malware helped conclude intended (D)TLS use can be permitted and malicious (D)TLS can be blocked.
• Malware agents cannot mimic (D)TLS profiles of several IoT devices and cannot keep up with the updates to (D)TLS profile.
Comments and suggestions are welcome
Collaboration to profile IoT devices
Request for WG adoption